

Advanced Achievement System

Author: Jake Aquilina

Email: realsoftgames@gmail.com

[Website](#) [Discord](#) [Youtube](#)

Modified Date: 23/02/2024

Source: C#

Version 2.2.1

Updated Documentation	
Google Docs	3
Video Tutorial	3
Short Tutorial	3
Full Tutorial	4
Code Example	8

Updated Documentation

[Google Docs](#)

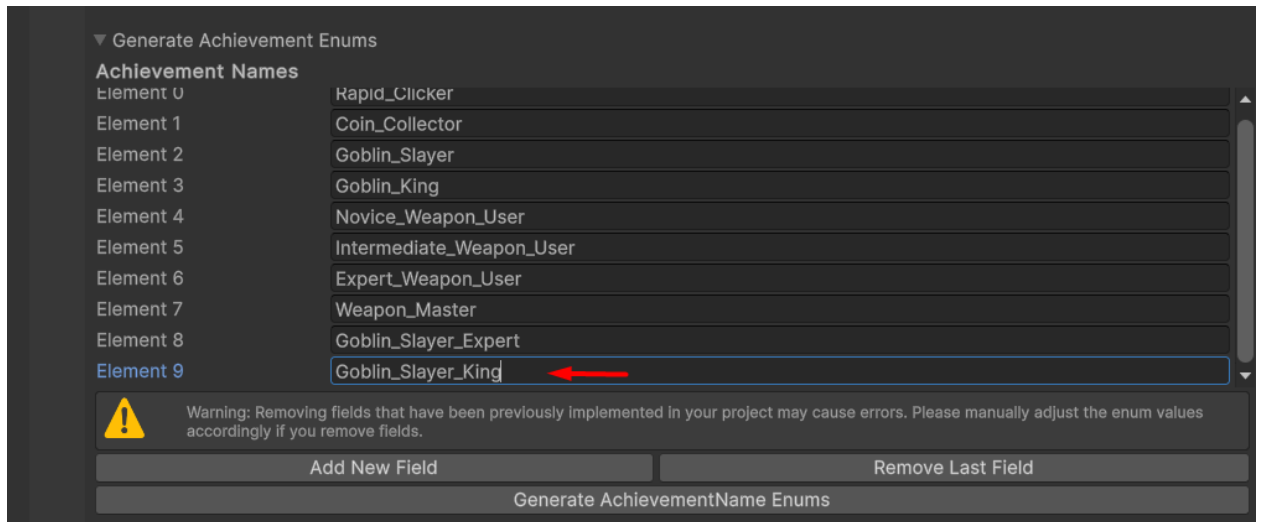
Video Tutorial

[YouTube](#) (Updated Video Tutorial Coming Soon, please check discord for updates or the online documentation)

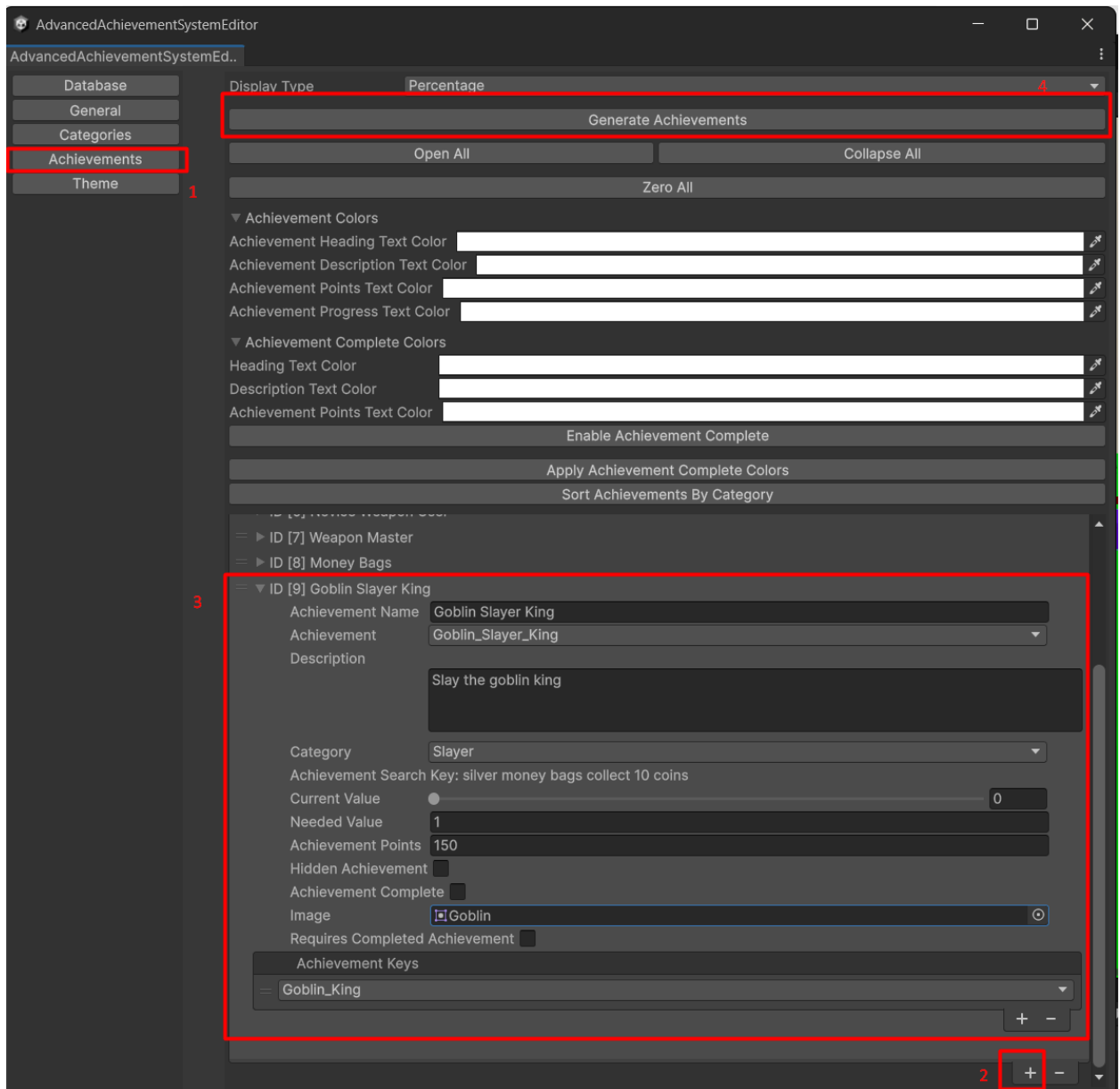
Short Tutorial

In this tutorial i will refer Advanced Achievement System as AAS

- 1) Import Advanced Achievement System into your project
- 2) open the Demo scene located at RealSoftGames > AdvancedAchievementSystem > Demo > Scenes
- 3) Navigate through toolbar at top of unity. Tools> RealSoft Games > Advanced Achievement System
- 4) Select a Theme pack and click install, select the demo database when prompted to
- 5) Enable the achievement system from the general tab, generate categories button and generate achievements button
- 6) Hit play to test the example achievements
- 7) Try adding a new achievement - Goblin King Slayer and rewarding the player with 100 silver when its complete, we will first add the achievement name under general tab

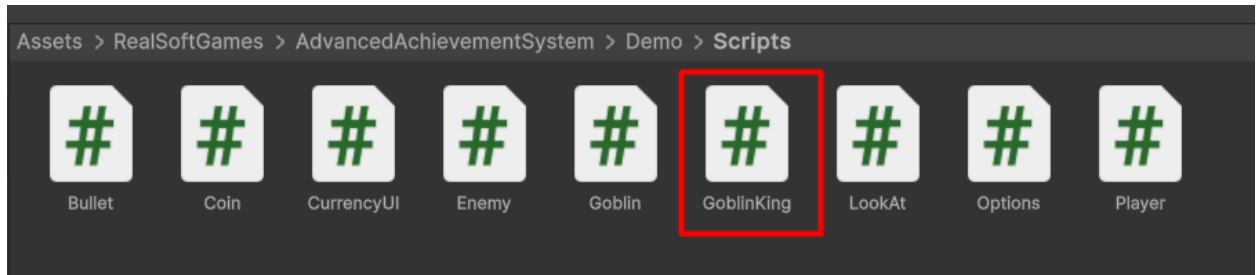


8) Next we will go to the achievement tab and create the achievement, follow the steps in the image below



Once you have filled out the achievement for the Goblin Slayer King, Click on the generate achievements to add it to the achievements list, after generating we can sort the achievements by category automatically or you can manually sort the achievements based on the array position, this is how the achievements will display in order in the achievement system.

9) Next we will open the GoblinKing.cs script located here



Then we can add this line here to the Death method
AchievementManager.Instance.UpdateAchievement(AchievementList
.Goblin_Slayer_King, OnAchievementCompleteCallback:
OnAchievementComplete);

So we essentially have access to the OnAchievementComplete for this method when we increment the current value, if the current value then reaches the total required value it will trigger the achievement complete callback which we can then access to grant the player rewards. Like so,

NOTE: There are 3 ways to increment achievements and 3 ways to obtain achievement complete or achievement updated events, either by Key or by directly updating the achievement. We will show each examples below

First Example is updating the achievement directly and accessing the callback directly from within the increment method call.

```
4 references
protected override void Death()
{
    AchievementManager.Instance.UpdateAchievementsByKey(AchievementKey.Goblin);

    AchievementManager.Instance.UpdateAchievement(AchievementList.Goblin_Slayer_King, OnAchievementCompleteCallback: OnAchievementComplete);

    base.Death();
}

4 references
private void OnAchievementComplete()
{
    Debug.Log("AchievementName Completed Goblin King Granting Reward of 100 silver");
    Player.AddCurrency(100);
}
```

The Second Example is updating all achievements by key and capturing any completed achievements regarding the call back for the specific achievement we are looking for.

```
4 references
protected override void Death()
{
    AchievementManager.Instance.UpdateAchievementsByKey(AchievementKey.Goblin);
    AchievementManager.Instance.UpdateAchievementsByKey(AchievementKey.Goblin_King, OnAchievementCompleteCallback: OnAchievementComplete);
    base.Death();
}

4 references
private void OnAchievementComplete(AchievementList achievement)
{
    if (achievement == AchievementList.Goblin_Slayer_King)
    {
        Debug.Log("AchievementName Completed Goblin King Granting Reward of 100 silver");
        Player.AddCurrency(100);
    }
}
```

The third method is by ensuring safe calling with multiple keys, In the first example it is still correct, however on the Goblin slayer achievements, Goblin key and Goblin king keys are accepted in updating the achievement progress, in example 1 we would be updating the Goblin achievements twice in a single call, where using the UpdateAchievementsByKeys allows us to update the achievements safely and only by 1 call by checking for any key provided

```
4 references
protected override void Death()
{
    //We update achievement by keys rather than individually by key here as the goblin king is a goblin and
    //We want the kings goblin progress to count towards other achievements but we do not want to call it 2 times,
    //So this call ensures an achievement is only updated 1 time

    AchievementManager.Instance.UpdateAchievementsByKeys(
        new AchievementKey[] {
            AchievementKey.Goblin,
            AchievementKey.Goblin_King
        },
        OnAchievementCompleteCallback: OnAchievementComplete);

    base.Death();
}
```

The fourth example is subscribing to the achievement directly and having the achievement system internally handle the OnAchievementUpdated and OnAchievementCompleted events

```

@ Unity Message | 0 references
private void Awake()
{
    AchievementManager.OnInitialized += OnInitialized;
}

1 reference
private void OnInitialized()
{
    Debug.Log("AchievementManager.Instance Has been initialized");
    AchievementManager.Instance.Database.Achievements.Find(i => i.Achievement == AchievementList.Goblin_King).OnComplete += OnAchievementComplete;
}

```

```

2 references
private void OnAchievementComplete()
{
    Debug.Log("Achievement Completed Goblin King");
    Player.AddCurrency(100);
}

```

The main difference is how the callback is handled in both OnAchievementComplete and OnAchievementUpdated callbacks, directly accessing the achievement we do not need to specify which achievement is obtaining a callback, where the second approach we need to specify which achievement callback we are looking for as multiple achievements could trigger OnAchievementComplete if the right conditions are met leading to granting multiple rewards, so here we need to specify which achievement has been completed in the callback method before granting the rewards.

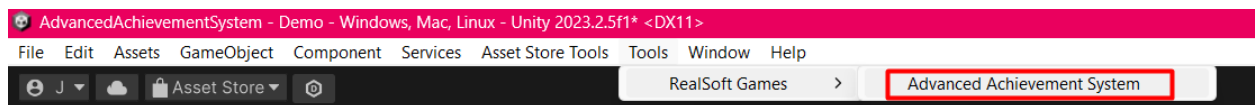
- 10) Choose one of the methods above for this tutorial, save the script and enter play mode and slay the Goblin King, press left Tab to open the achievement menu and look for the goblin king slayer achievement, it will be completed and in the top left of your screen you should have acquired the reward for slaying the goblin king, also note only 1 of the rewards will be granted as there are now 2 Goblin slayer achievements.

Full Tutorial

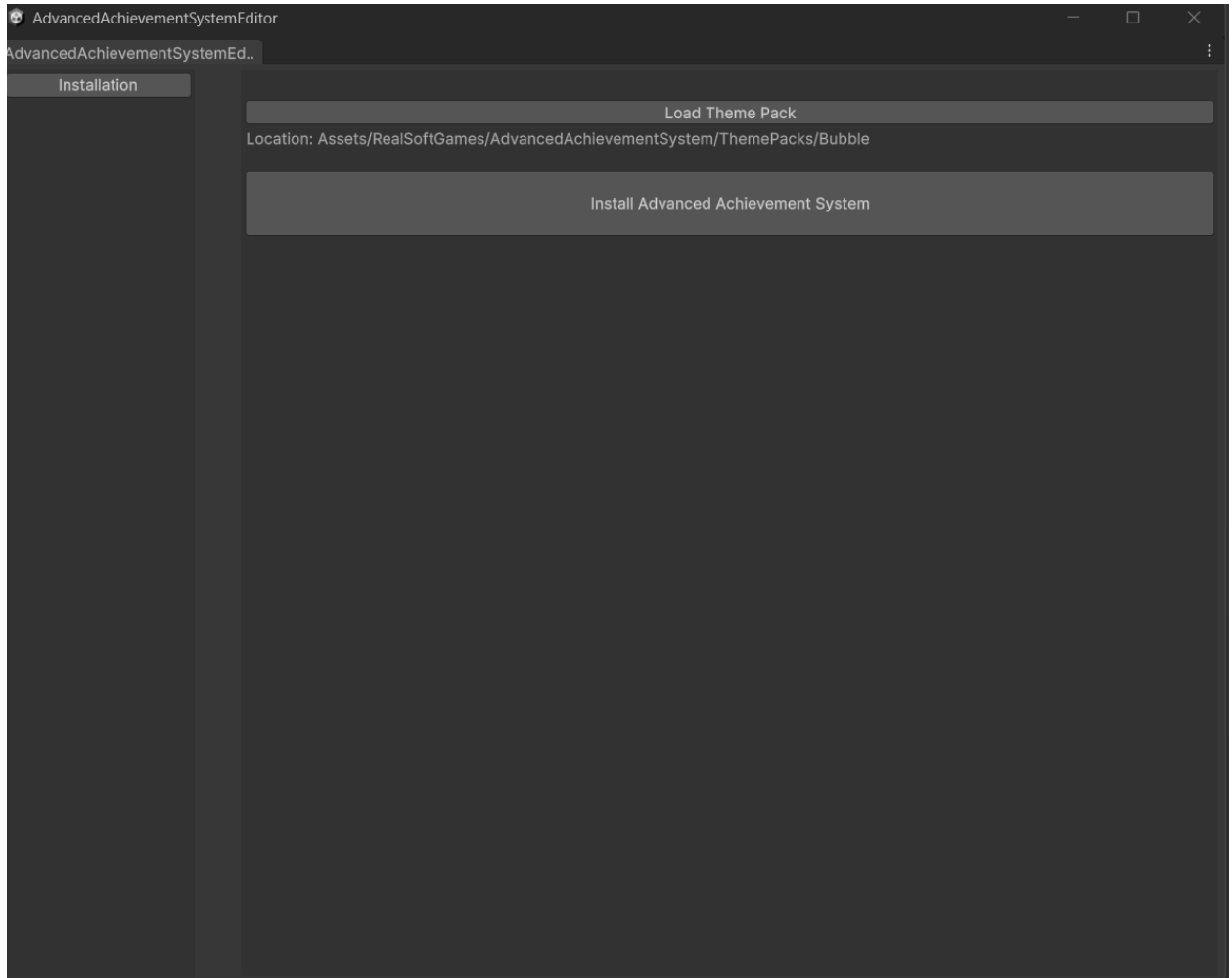
In this tutorial i will refer Advanced Achievement System as AAS.

In this tutorial we will go over how to import and use AAS in your own game.

- 1) Import Advanced Achievement System into your project
- 2) In the navigation bar open Tools > RealSoft Games > Advanced Achievement System

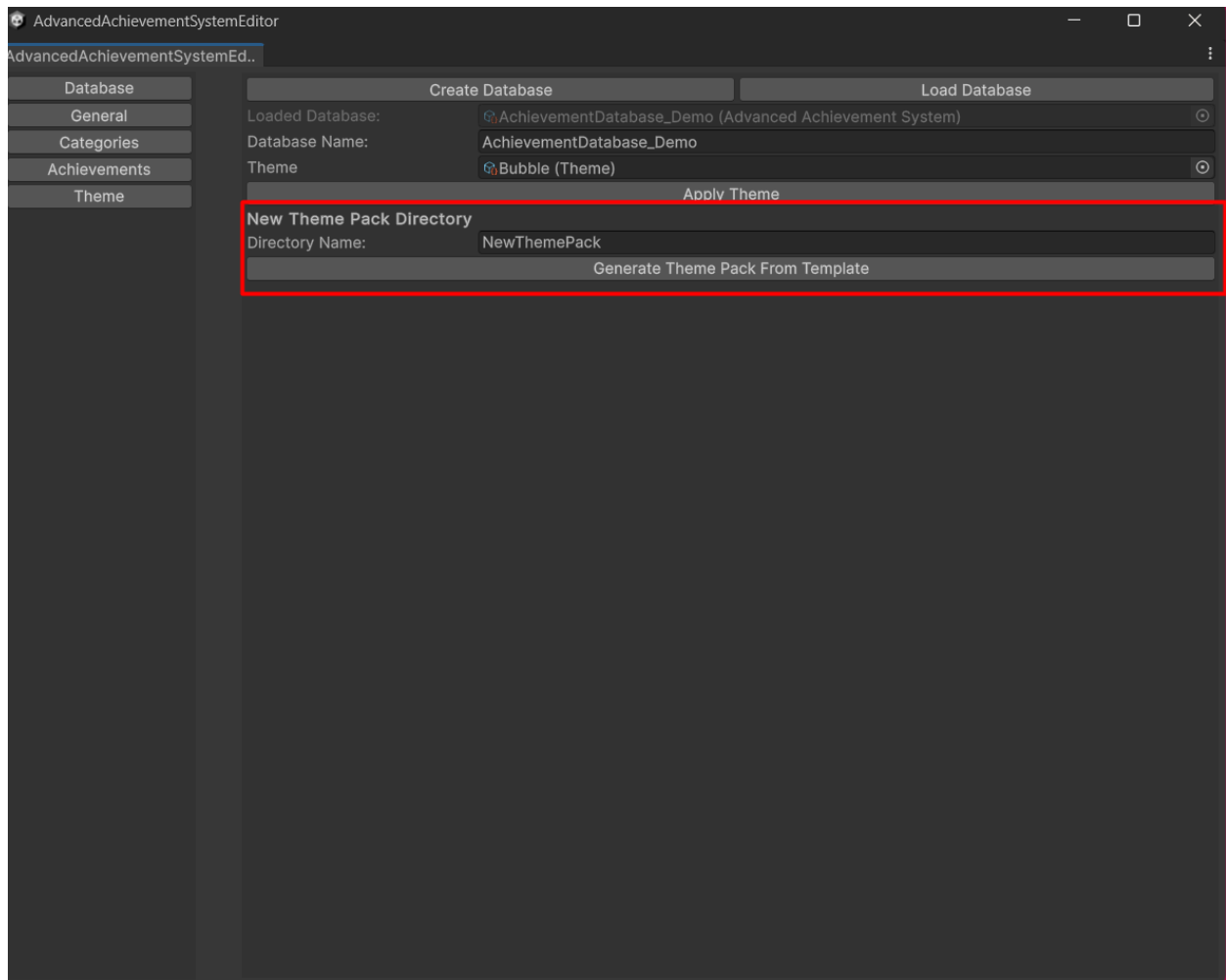


- 3) Install a pre made theme pack by clicking Load Theme Pack and navigating to
Assets\RealSoftGames\AdvancedAchievementSystem\ThemePacks
- Select a theme pack directory to load (each of the theme packs have already been adjusted for their theme, you can delete the instance in the hierarchy later to install a different theme pack, this will not effect the database as the theme pack is just the Graphical side displaying to the end user)



- 4) When you click install a prompt will ask you to load or create a new achievement database, for this example we will just load the demo database which has some pre filled fields which can be removed later
 - 5) If you want to apply your own theme, this can be easily done in this step, otherwise skip step 5
- Applying your own theme is easy, after installing a pre made theme pack, give your theme pack a custom name for example RSGTheme

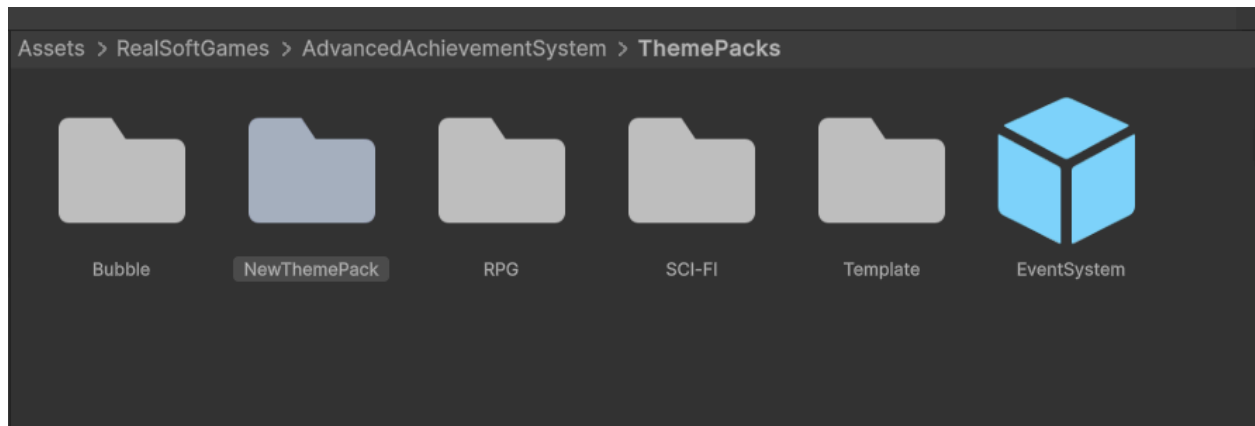
then click create them pack from template (this will use the bubble them by default which is applied to the Template directory)



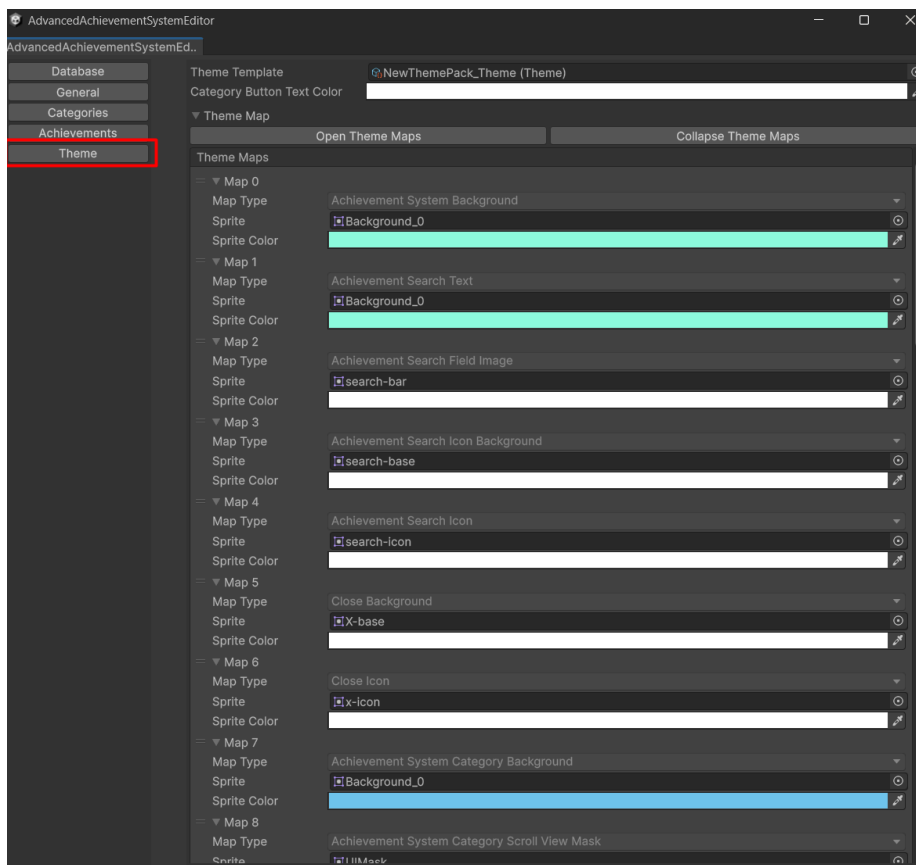
Generating a new theme pack allows you to add your own custom graphics to the achievement system without having to navigate through the Hierarchy and applying graphics individually. We can handle this all through the editor.

A new directory will be created in the ThemePacks directory with your custom theme pack. Doing this will automatically delete the instantiated

instance of the achievement system and create a new instance with the newly created theme pack (The template utilises the Bubble them by default for the template).



To Change the graphics of the template head over to the Theme tab and apply your theme graphics here and adjust the colors accordingly



6) Head over to the General Tab, and in here we will setup all of the categories, achievements and achievement keys they will be laid out in this format for each of the generation steps.

Categories - This is where we will generate the category enum values, give them descriptive names, these must be enum compatible

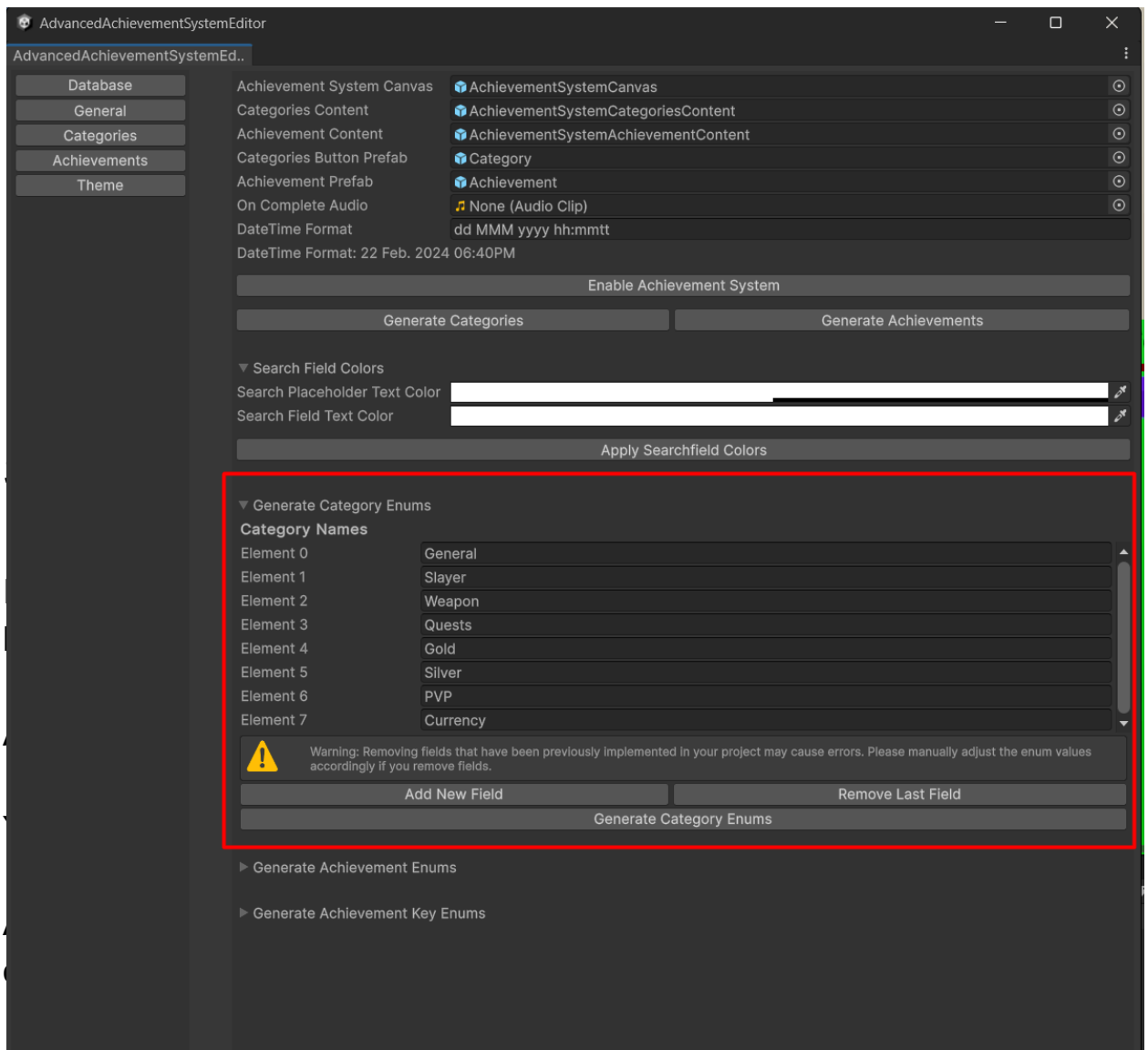
Achievements - Each achievement will be handled through an Enum value, this is one of the key improvements from the old string system. This ensures we don't make typos with string values and is much simpler to use with the new editor window.

Achievement keys - This section is how you will handle achievements in your code, for example you may have several achievements listening for the same output. We can use these 2 achievements for example.

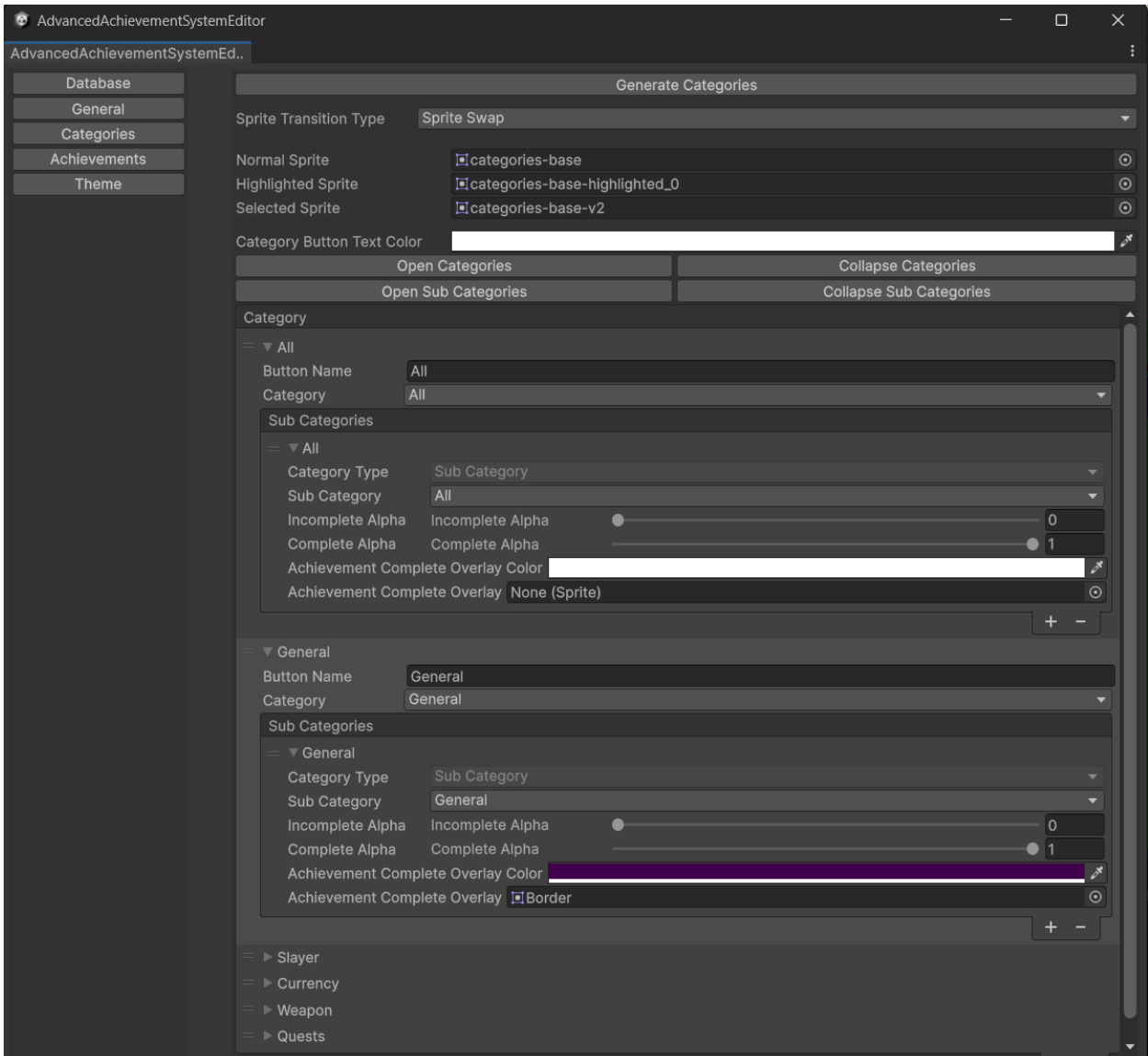
Novice weapon user and Intermediate weapon user, both of these achievements are both listening for an achievement key called Weapon.

From the general tab you can also Enable and disable the achievement system in the editor for making modifications easily and visualising the changes you make.

You are able to change the search field text colors from the general tab also and even generate the categories and achievements from here in one place

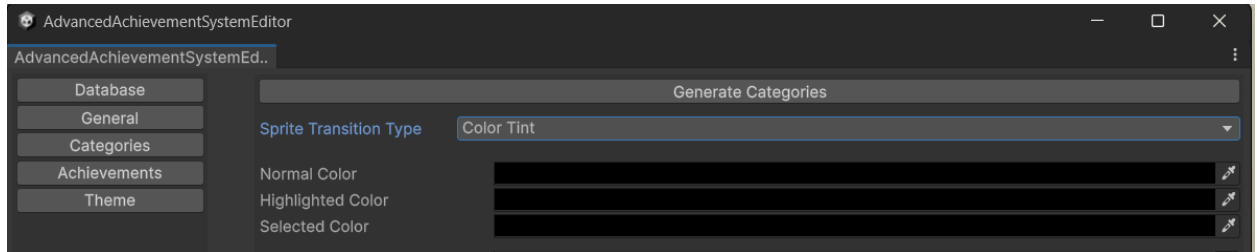


7) After setting up your categories, achievements and achievement keys to use we will start by setting up our Categories. Head over to the Categories Tab



Setting the Sprite transition type to Sprite will allow you to switch the sprites out from normal, highlighted and selected (The text will be generated based on the Category (Button Name))

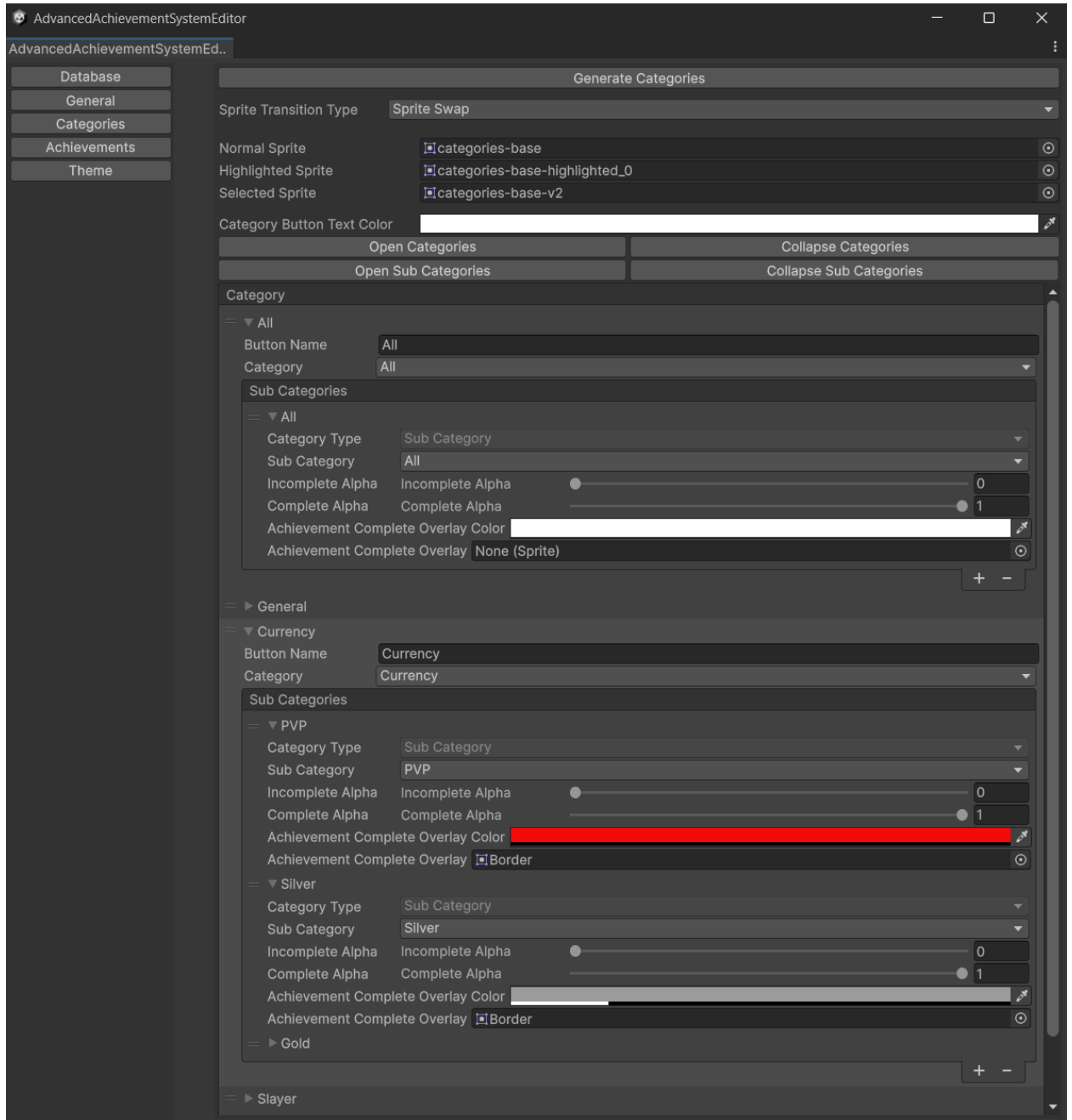
You can also use a single sprite and color swap between also with the Color swap selection



- 8) The All category type is the default for displaying unfiltered category selection, where here we can add primary categories and then add sub categories (as we created the categories in the previous step in the General tab)
- Here we have 2 types of categories, Primary and Sub category, The Primary category will be the Button which controls which sub categories are filtered through the selection process in game. For example Currency contains Silver, Gold, PVP selecting the Currency tab will display all sub categories assigned to it. As well as any primary categories assigned by the achievements.

The achievement complete overlay is a border around your achievement style which helps indicate an achievement as being complete, you are able to assign the border and color tint based on each of the sub category types

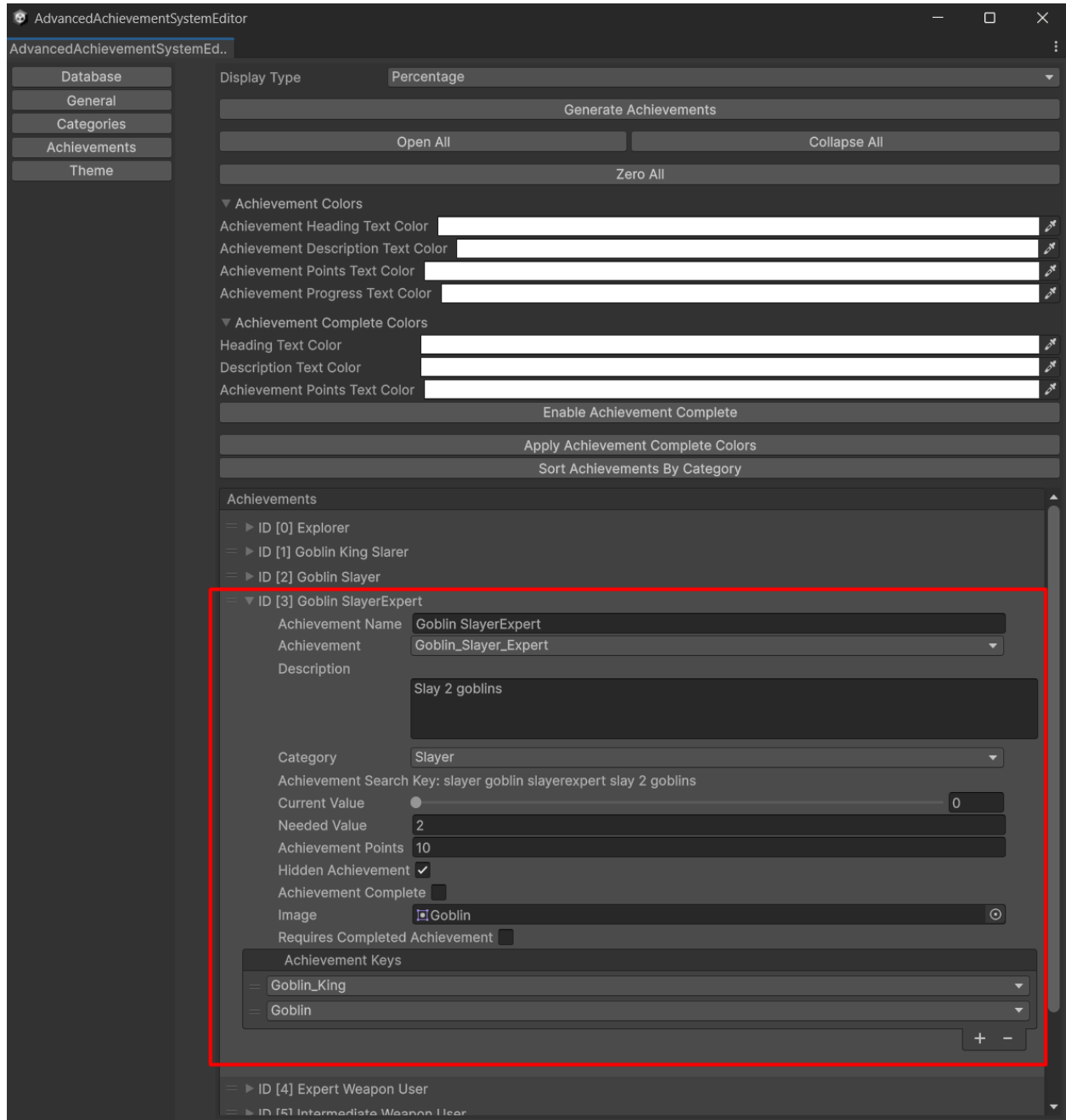
Now that we have our categories setup we can click on generate categories at the top and this will generate all the category buttons in your achievement system.



9) Now over to the achievement tab, here we can assign the text color to the achievements as well as the achievement system pretty easily for a little customisation, to refine the customisation you will need to open the prefab and adjust the sizing and positioning accordingly

- Assign the achievement name to display in the achievement, then assign the achievement enum which we created earlier in the General tab for this particular achievement.
- Assign it a category, this is used by the category buttons to filter what achievements should be displayed.
- Current value should always be set to 0 unless for some reason you want them to start with a value greater than 0.
- Needed value is how many points are needed to complete this achievement based on the current value
- Description is for the description of the achievement to show up in the achievement manager
- Achievement points is a display of how many achievement points this achievement is worth when its completed
- Assign an Icon for the achievement
- Is the achievement hidden by default (does not display in the achievement menu, hence the hidden achievement. And only displays if it is completed)
- Requires completed achievement, This check will allow you to assign another achievement to this achievement so you can chain the completion. For example slay 10 goblins, slay 50 goblins slay 500 goblins, if you dont chain them and they all contain the goblin achievement key progress will go towards all 3 achievements at the same time, where if you assign them as required achievements they will only assign progress when the prior is complete.

- Achievement keys is how we will assign progress to the achievements via code.



Code Example

Update Achievements directly with call backs, or update achievements through the Achievements Key

```
void AchievementManager.UpdateAchievementsByKey(AchievementKey key, [Action OnUpdateCallback = null], [Action OnAchievementCompleteCallback = null])  
Update AchievementName(s) by the key
```

```
Unity Message | 0 references  
private void OnTriggerEnter(Collider col)  
{  
    if (col.name == "Player")  
    {  
        Player.AddCurrency(1);  
        AchievementManager.Instance.UpdateAchievement(AchievementList.Coin_Collector);  
        AchievementManager.Instance.UpdateAchievementsByKey(AchievementKey.Coin);  
        Destroy(gameObject);  
    }  
}
```

```
protected override void Death()  
{  
    AchievementManager.Instance.UpdateAchievementsByKey(AchievementKey.Goblin_King);  
    AchievementManager.Instance.UpdateAchievementsByKey(AchievementKey.Goblin, null, null);  
    base.Death();  
    if (AchievementManager.Instance != null)  
        AchievementManager.Instance.Database.Achievements.Find(i => i.Achievement == AchievementList.Goblin_King).OnComplete -= OnAchievementComplete;  
}
```

Creating Achievement Rewards

We First Start by creating a new script and inheriting from AchievementReward (Note dont forget to add the namespace)

```
using RealSoftGames.AdvancedAchievementSystem;
```

(Reference for the Base Object)

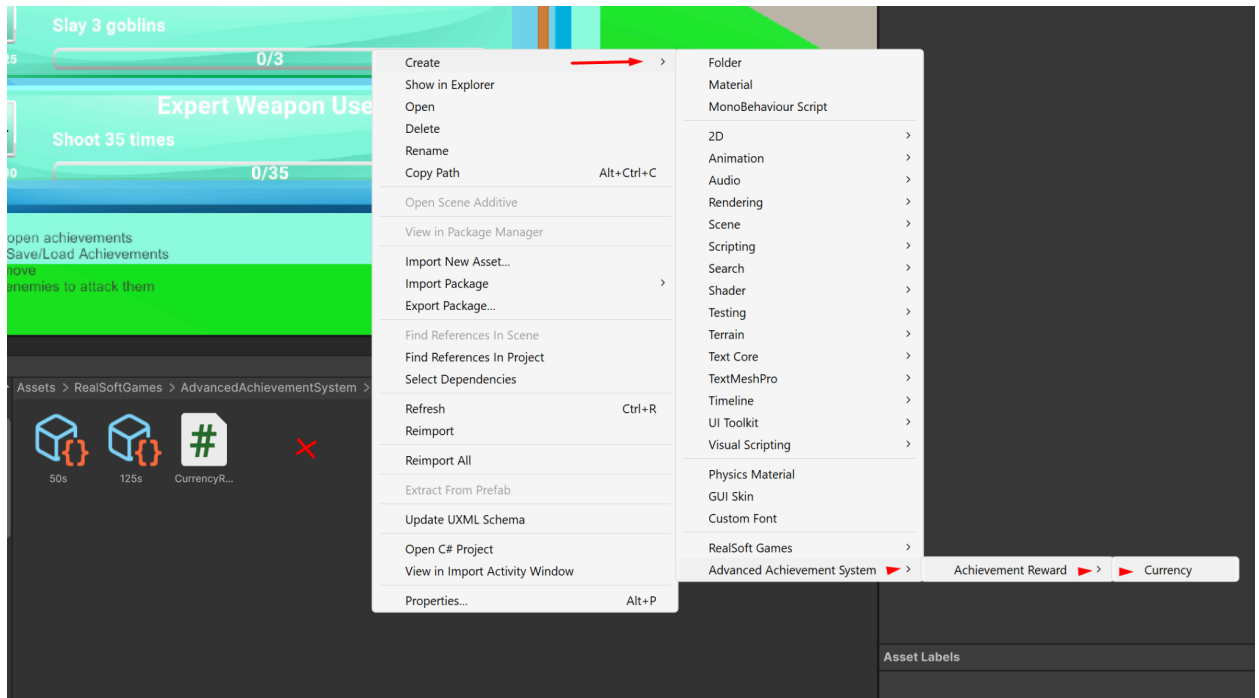
```
namespace RealSoftGames.AdvancedAchievementSystem
{
    5 references | 0 assets
    public abstract class AchievementReward : ScriptableObject
    {
        1 reference
        [SerializeField] Sprite rewardIcon;
        1 reference
        [SerializeField] int rewardAmount = 1;
        2 references
        public abstract void Execute();

        0 references
        public Sprite RewardIcon { get => rewardIcon; }
        1 reference
        public int RewardAmount { get => rewardAmount; }
    }
}
```

Assign the fileName; This is what the default name of this item will be when its initially created, Also add a Menu name for when you right click in the project directory you can create a new object of this type.

```
[CreateAssetMenu(fileName = "CurrencyReward", menuName = "Advanced Achievement System/Achievement Reward/Currency", order = 1)]
0 references | 2 assets
public class CurrencyReward : AchievementReward
{
    2 references
    public override void Execute()
    {
        Player.AddCurrency(RewardAmount);
    }
}
```

By right Clicking in the project Directory you can now create a new object of the type you just created, you can create multiple types and configure them as you desire, for example generating random rewards from a loot box. Execute is the main entry point and is only triggered once when the achievement is first completed. It is not re triggered off of a save file.



You can then drag and drop the newly created achievement reward into the achievement you wish to grant the reward upon completion.

